ESPECIFICACIÓN DEL PROGRAMA

INTRODUCCIÓN

- Se parte de:
  - especificaciones de requerimientos (hechas por el cliente)
  - plan del proyecto
  - estudio de viabilidad económica

- La comprensión de los requerimientos es fundamental

- Básicamente es una tarea de análisis que comprende:
  - descubrimiento
  - modelado
  - refinamiento
  - especificación

- Posibilidad muy alta de problemas de comunicación:
  - mala interpretación
  - ambigüedad
  - desinformación

ESPECIFICACIÓN DEL PROGRAMA

INTRODUCCIÓN

- La especificación ha de definir:
  - modelo del dominio
  - funcionalidad
  - rendimientos
  - interfícies con otros sistemas
  - restricciones del diseño

- Las tareas son:
  - identificar el problema
  - evaluar y sintetizar la solución del “qué” y no del “cómo”
  - modelado
  - especificación
  - revisión
  - manual de usuario (¡!)
  - revisión del plan del proyecto

ESPECIFICACIÓN DEL PROGRAMA

FORMAS DE ESPECIFICACIÓN

- Diagramas de clases
- Diagramas de casos de uso
- Diagramas jerárquicos
- Diagramas de flujo de datos (DFD)
- Diagramas de estado, de secuencia, de colaboración, ...
- Texto libre
  - 70% de los errores
  - Buena redacción (comprensible)
  - Meyer definió los 7 “pecados” del especificador
    - Ruido – información no necesaria o redundante
    - Silencio – falta de información
    - Sobre-especificación – detallar más de lo necesario
    - Contradicción – entre partes de la especificación
    - Ambigüedad – por falta de precisión, por ejemplo por el uso de sinónimos y homónimos
    - Referencias hacia delante – utilizar términos no definidos previamente
    - Remordimiento – hacer precisiones o introducir restricciones sobre declaraciones anteriores
- Glosario

Proceso cíclico entre documentos
Todos los elementos de una especificación han de ser **consistentes** entre sí

ESPECIFICACIÓN DEL PROGRAMA

AMPLIACIÓN DEL ENUNCIADO

- Contenido:
  - Qué se espera del sistema
  - Qué funcionalidades tendrá
  - En qué entorno se trabajará
  - Qué restricciones ha de cumplir

- 2 clases de funcionalidades:
  - De usuario
    - Explícitas
    - Implícitas
  - De sistema
    - Requerimientos no funcionales

- 2 tipos de requerimientos:
  - Obligatorios
  - Opcionales

ESPECIFICACIÓN DEL PROGRAMA

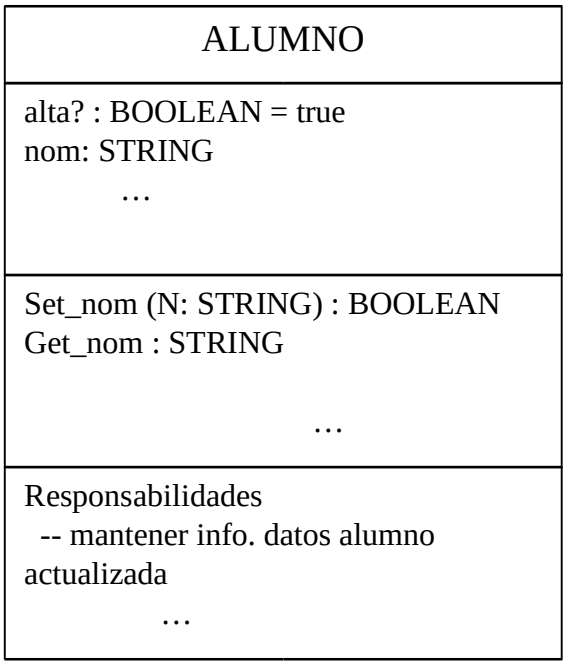
DOMINIO DE LA INFORMACIÓN

- Toda aplicación informática es de procesamiento de datos
- Procesar información comprende 3 aspectos:
  - Contenido
  - Estructura
  - Flujo
- La calidad y precisión del modelo permite:
  - Extraer mejor la funcionalidad que se quiere
  - Hacer esta funcionalidad más dúctil al cambio
- Las buenas abstracciones son reutilizables

Contenido

- Son los datos individuales y su agrupación en entidades reales de nuestro dominio
  - Ej: Sistema de catalogación bibliográfica
    se compone de entradas con autor, título, etc.
- Cada entidad corresponderá a una clase
- Cada dato a un atributo de la clase
- Se describen con una plantilla por clase y atributo

DOMINIO DE LA INFORMACIÓN: CONTENIDO



- Extracción de las clases:
  - Primera aproximación: modelizar vocabulario del sistema: extraer nombres (⇐ clases) y adjetivos (⇐ atributos)

    ⇩

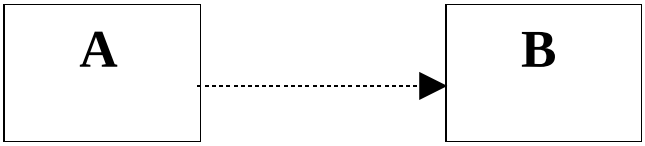
    Refinamiento: abstracciones con entidad propia
  - Para cada abstracción, identificar un conjunto de responsabilidades ⇨ atribuirle propiedades (y servicios)
  - Asegurarse de mantener un buen equilibrio entre las responsabilidades de las clases:
    - Clases demasiado grandes ⇨ modelos difíciles de modificar y poco reusables

- Clases demasiado pequeñas más abstracciones de las que se pueden manejar y entender razonablemente

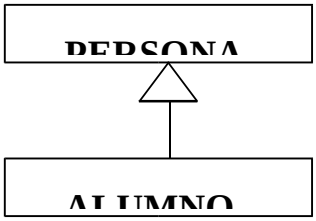
DOMINIO DE LA INFORMACIÓN: ESTRUCTURA (I)

Tipos de Relaciones:

1. Dependencia: Relación de uso ⇨

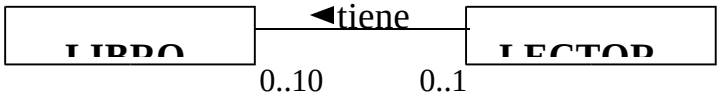


2. Generalización: Relación de herencia (“is-a”)

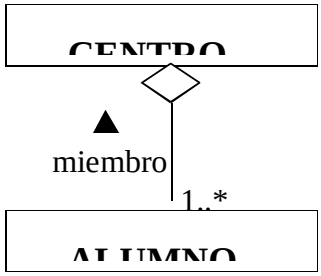


3. Asociación o instancia: Relación estructural

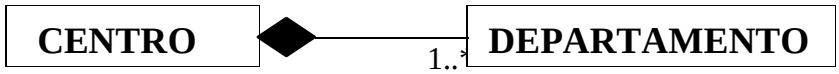
3.1. Asociación pura



3.2. Agregación: La clase “parte” tiene sentido sin estar asociada a la clase “todo” concreta



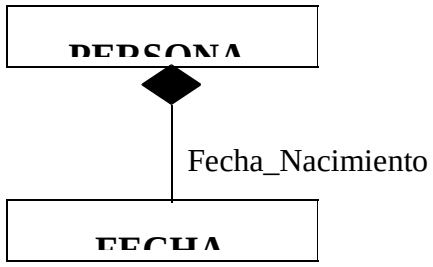
3.3. Composición: La “parte” no tiene sentido sin el “todo” concreto



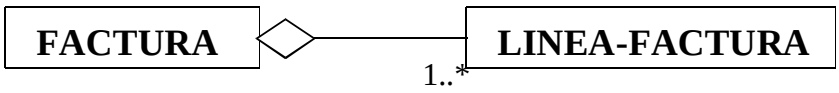
DOMINIO DE LA INFORMACIÓN: ESTRUCTURA (II)

Ejemplos:

- Agregación clarísima:



- Duda asociación-agregación:



Modelización de relaciones estructurales:

- Para todo par de clases, si desde un objeto de una clase se necesita ver objetos de otra ⇨ data-driven view
- Para todo par de clases, si objetos de una clase necesitan interactuar con objetos de otra (no como parámetros de una operación) ⇨ behaviour-driven view
- Para toda asociación, si una de las clases es estructural u organizativamente un “todo” en comparación con la clase del otro extremo ⇨ agregación/composición
- Atribuir a la relación cardinalidad, role-names, visibilidad

ESPECIFICACIÓN DEL PROGRAMA

REQUERIMIENTOS NO FUNCIONALES

- Son requerimientos no relacionados con funciones de usuario

- Han de ser medibles:

  - “tiempo de respuesta < 3ms” versus “tiempo de respuesta rápido”
  - “se aprende en 3 días por un administrativo” versus “aprendizaje fácil”

- Se refieren a:

  - Interfície de usuario y factores humanos
  - Documentación
  - Consideraciones de hardware
  - Características de rendimiento
  - Tratamiento de errores y condiciones extremas
  - Interfície del sistema
  - Factores de calidad
  - Modificaciones al sistema
  - Temas de seguridad
  - Entorno de desarrollo

  [incluir sólo los que aporten información adicional!]